Zoea

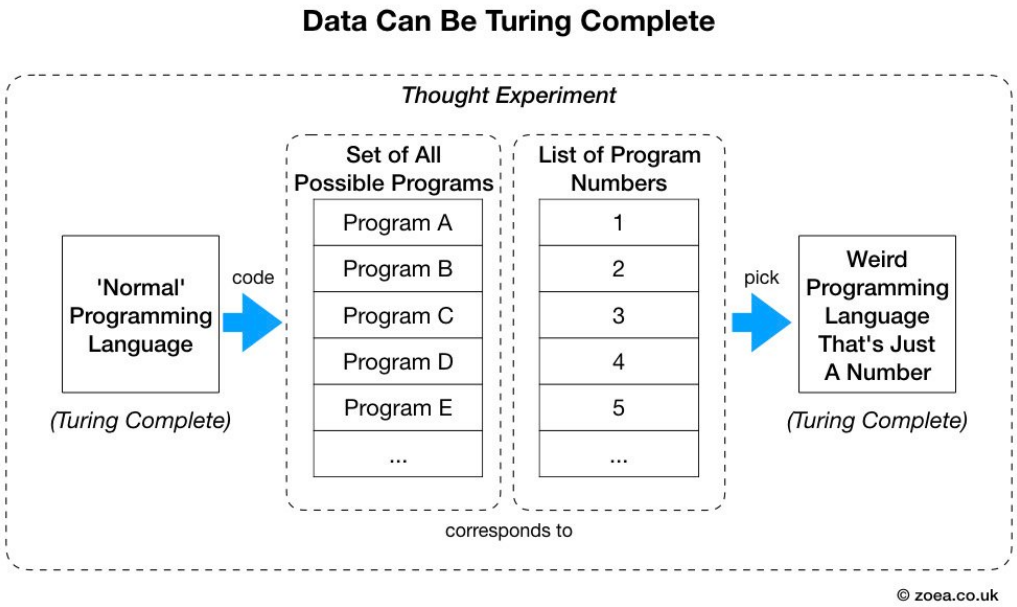
Research Note 11

Can Data be Turing Complete?

Edward McDaid & Sarah McDaid

23 Feb 2021

It isn't difficult to show that the Zoea programming language is 'Turing complete'. However, this raises some interesting questions about the nature of programming languages.



Zoea is a simple to use programming language. It is simple because it eschews many of the elements that people normally associate with programming languages such as variables, control structures, functions and so on. Instead a Zoea program is mostly just examples of input and output data - much like a set of test cases. Nevertheless by any

definition Zoea is definitely a programming language - it allows a person to formally specify the behaviour of a given program which it then generates.

Despite its simplicity Zoea is also what is called ‘Turing complete’. In every day English this basically means that Zoea can - at least in theory - be used to write any program that you can write in any other programming language.

How can we show that Zoea is Turing complete? One way to do so would be to use Zoea to create a universal Turing machine. This is certainly possible. An easier way is to use Zoea to create something else that is also Turing complete such as a string rewriting system or Conway’s Game of Life.

It is a trivial matter to define a string rewriting system using Zoea. Rewrite rules can be expressed directly as test case inputs and outputs. In addition any number of example reductions involving any combination of rewrite rules can also be provided as test cases. The rewrite rules and the example reductions are all equally valid test cases for any given rewrite system.

Being Turing complete is good news for Zoea and in itself it wouldn’t be very remarkable except for the fact that Zoea programs are basically just data. You see data is not supposed to be Turing complete.

The whole idea of Turing completeness grew from the question of what algorithms can be expressed in a given programming language. This is all fine so long as we stick to programming languages where we explicitly specify the steps of algorithms. The thing is that in Zoea we don’t do that.

Here’s a simple thought experiment that shows how some different data can also be Turing complete. Suppose we create a list with every possible program in a given programming language starting with the shortest and we assign each program a unique number. We can now specify and potentially run any given program simply by referencing its associated number. This is equivalent to creating a new programming language where each program is specified by a different integer. In this scenario the list of possible program numbers is also Turing complete as there is a number for every possible program including at least

one for a universal Turing machine. (Just to be clear this approach isn't how Zoea works internally.)

So what's going on here? Maybe the computability theory definition of a programming language is a little narrow. It is entirely possible for a system of data manipulation rules to be Turing complete even if we have no direct knowledge of what those rules are. It turns out that quite a few complex systems happen to be Turing complete even though they weren't designed to be. Maybe computability in general is just a more common natural phenomena than we assume.

It is also clear that the interpretation that is applied to data is important. The list of numbers in our thought experiment is only Turing complete if we interpret a number as a program. In a different context the same number would be just a value.

Thankfully it will be up to the experts to sort all this out. Regardless of how they do Zoea will still be a simple programming language - that also happens to be Turing complete.

Learn more at **zoea.co.uk**